Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended) A method in a data processing system [[for]] of presenting coverage data for code, the method comprising:

obtaining the coverage data containing instruction access indicators associated with the code, wherein each instruction access indicator is associated with a different portion of the code, and wherein each instruction access indicator is initialized as being unset prior to execution of its associated code portion;

identifying ~~particular~~ instruction access indicators that have been set by a processor in the data processing system in response to execution of the code by the processor to form set instruction access indicators, wherein each set instruction access indicator is associated with [[a]] an executed portion of the code; and

generating a presentation for the coverage data, wherein [[the]] each set instruction access ~~indicators are~~ indicator is identified in the presentation.

2. (Currently amended) The method of claim 1 further comprising:

identifying unset instruction access indicators that have remained unset during the execution of the code by the processor, wherein [[the]] each unset instruction access indicator is associated with an unexecuted portion of the code, and wherein each unset instruction access ~~indicators are~~ indicator is identified in the presentation.

3. (Original) The method of claim 2, wherein the set instruction access indicators are identified in the presentation using a first color and wherein the unset instruction access indicators are identified in the presentation using a second color.

4. (Original) The method of claim 2, wherein the set instruction access indicators are identified in the presentation using a graphical indicator and wherein the unset instruction access indicators are identified in the presentation using the graphical indicator.

5. (Original) The method of claim 2, wherein the generating step is performed in response to an event.

6. (Original) The method of claim 5, wherein the event is at least one of a completion of the execution of the code, expiration of a time, and the execution of a selected type of instruction in the code.

7. (Original) The method of claim 1, wherein the portion of the code is a single instruction in the code and wherein every instruction in the code is associated with a different instruction access indicator.

8. (Original) The method of claim 1, wherein the portion of the code is a subroutine in the code.

9. (Original) The method of claim 1, wherein the portion of the code is a branch instruction in the code.

10. (Currently amended) A data processing system ~~for presenting~~ that presents coverage data for code, the data processing system comprising:

a processor for processing computer instructions;

a memory coupled to the processor using a bus, the memory comprising:

obtaining means ~~for obtaining~~ that obtains the coverage data containing instruction access indicators associated with the code, wherein each instruction access indicator is associated with a different portion of the code, and wherein each instruction access indicator is initialized as being unset prior to execution of its associated code portion;

identifying means ~~for identifying particular~~ that identifies instruction access indicators that have been set by a processor in the data processing system in response to execution of the code by the processor to form set instruction access indicators, wherein each set instruction access indicator is associated with [[a]] an executed portion of the code; and

generating means ~~for generating~~ that generates a presentation for the coverage data, wherein [[the]] each set instruction access ~~indicators are~~ indicator is identified in the presentation.

11. (Currently amended) The data processing system of claim 10, wherein the identifying means is a first identifying means and further comprising:

second identifying means ~~for identifying~~ that identifies unset instruction access indicators that have remained unset during the execution of the code by the processor; wherein [[the]] each unset instruction access indicator is associated with an unexecuted portion of the code, and wherein each unset instruction access ~~indicators are~~ indicator is identified in the presentation.

12. (Original) The data processing system of claim 11, wherein the set instruction access indicators are identified in the presentation using a first color and wherein the unset instruction access indicators are identified in the presentation using a second color.

13. (Original) The data processing system of claim 11, wherein the set instruction access indicators are identified in the presentation using a graphical indicator and wherein the unset instruction access indicators are identified in the presentation using the graphical indicator.

14. (Original) The data processing system of claim 11, wherein the generating means is performed in response to an event.

15. (Original) The data processing system of claim 14, wherein the event is at least one of a completion of the execution of the code, expiration of a time, and the execution of a selected type of instruction in the code.

16. (Original) The data processing system of claim 11, wherein the portion of the code is a single instruction in the code and wherein every instruction in the code is associated with a different instruction access indicator.

17. (Original) The data processing system of claim 11, wherein the portion of the code is a subroutine in the code.

18. (Original) The data processing system of claim 11, wherein the portion of the code is a branch instruction in the code.

19. (Currently amended) A computer program product in a recordable-type computer readable medium ~~for presenting~~ that presents coverage data for code, the computer program product comprising:

first instructions ~~for obtaining~~ that obtain the coverage data containing instruction access indicators associated with the code, wherein each instruction access indicator is associated with a different portion of the code, and wherein each instruction access indicator is initialized as being unset prior to execution of its associated code portion;

second instructions ~~for identifying particular~~ that identify instruction access indicators that have been set by a processor in the data processing system in response to execution of the code by the processor to form set instruction access indicators, wherein each set instruction access indicator is associated with [[a]] an executed portion of the code; and

third instructions ~~for generating~~ that generate a presentation for the coverage data, wherein [[the]]each set instruction access ~~indicators are~~ indicator is identified in the presentation.

20. (Currently amended) The computer program product of claim 19 further comprising:

fourth instructions ~~for identifying~~ that identify unset instruction access indicators that have remained unset during the execution of the code by the processor; wherein [[the]] each unset instruction access indicator is associated with an unexecuted portion of the code, and wherein each unset instruction access ~~indicators are~~ indicator is identified in the presentation.

21. (Original) The computer program product of claim 20, wherein the set instruction access indicators are identified in the presentation using a first color and wherein the unset instruction access indicators are identified in the presentation using a second color.

22. (Original) The computer program product of claim 20, wherein the set instruction access indicators are identified in the presentation using a graphical indicator and wherein the unset instruction access indicators are identified in the presentation using the graphical indicator.

23. (Original) The computer program product of claim 20, wherein the third instructions is performed in response to an event.

24. (Original) The computer program product of claim 23, wherein the event is at least one of a completion of the execution of the code, expiration of a time, and the execution of a selected type of instruction in the code.